Assessing the performance of computerized tools for inventive design: insights from unsatisfactory outcomes

Assessing the performance of computerized tools for inventive design: insights from unsatisfactory outcomes

N. Becattini[1], Y. Borgianni[2], G. Cascini[1], F. Rotini[2]

Abstract Computers actually support, almost automatically, routine tasks such as those related to the optimization in design. Besides, the scientific community shows a growing interest in developing computer systems to aid non-routine tasks as a key to enhance individuals' creativity and innovation potential. In such a context, several attempts have been made to create tools based on the TRIZ logic to support inventive problem solving; some of them have been commercialized since decades, but still there is no established paradigm and all of them suffer from several limitations. So far the analysis of those limitations has been focused on the structure and on the nominal features of the software tools, while no in-depth and systematic investigation has been made to identify the reasons behind the partial failure of the existing systems. This paper proposes a set of general criteria to perform the evaluation of computerized tools supporting inventive design and reports an exemplary application, through protocol analysis, to the dialogue-based computerized algorithm for problem analysis, published by the authors in the past.

1 Introduction

Computer support to the product development process is widespread in the industry since the last decades. However, current artificial intelligence resources allow to automate just routine activities, such as those involved in optimization problems, i.e. when computers are used to choose the most appropriate value of a predefined set of variables, but no significant qualitative modifications are expected. Typically, those tasks emerge in the last phases of the design process, when the range of possible choices is limited to the details of the system.
On the other hand, in the last years, the scientific literature is collecting a growing number of contributions about the introduction and the development of computerized systems for supporting the early stages of product development cycle, namely

[1] N. Becattini, G. Cascini (✉)
Politecnico di Milano – Dipartimento di Meccanica, Via La Masa 1, 20156 Milano, Italy
e-mail: niccolo.becattini@kaemart.it, gaetano.cascini@polimi.it

[2] Y. Borgianni, F. Rotini
Università degli Studi di Firenze – Dipartimento di Meccanica e Tecnologie Ind.li
Via di Santa Marta 3, 50139 Firenze, Italy
e-mail: yuri.borgianni@unifi.it, federico.rotini@unifi.it

the design stages where it is required to solve inventive problems and where creativity plays a paramount importance [12]. Yet, cognitive processes involved in those design stages, e.g. ideas association, analogies, concept blending, are still activities completely in charge of human beings and computers role is barely aimed at supporting and fostering creativity either through a more efficient visualization of the mental model under study, or by guiding the reasoning path according to standardized strategies [14].

According to Funke and Frensch [9], problem solving is the most complex intellectual activity, because it deals with a number of tough characteristics as the presence of a large set of design variables mutually tangled, the need to satisfy multiple goals (politely) and the lack of clarity. With reference to this subject, Simon [15] suggested the distinction between ill and well-structured problems in design, according to a specific set of characteristics. An emerging branch of study is constituted by Computer-Aided Innovation (CAI). CAI Systems, often based on TRIZ theory, aim at supporting problem setting by guiding the designer to the formalization of a problem in terms of contradictions. TRIZ potential contribution to enhance industrial innovation has been largely acknowledged, as for example in [16]. Besides, all the existing software applications show several lacks in terms of real usability and usefulness.

In order to face this limitation, the authors have firstly dedicated proper efforts to the identification of the requirements a computer-aided inventive problem solving tool should satisfy; then they have developed a system implemented in a dialogue-based framework, namely OPEN-IT, which supports the problem setting phase by structuring the information according to TRIZ logic and fosters a learning-by-doing process (e.g. by teaching how to recognize the relevant aspects of a problematic situation) [4]. Two different groups of testers, holding a degree in Mechanical Engineering and with a scarce exposition to TRIZ theory, have demonstrated that the algorithm gives a significant contribution in the analysis of an inventive problem as presented in [4] and [3]. However, such experimental activity also highlighted that some testers still encounter difficulties during the definition of the problem characteristics, potentially leading to unsuccessful analyses.

An immediate conclusion arises from this experience: either it is hard to properly identify the requirements of a computerized tool for supporting inventive design, or it is harder than expected defining an algorithm capable to satisfy those requirements. None of the papers in the TRIZ or in the CAI literature provide relevant directions to address this dichotomy; thus, it is a relevant matter of investigation the definition of some general criteria and practices to assess the functionality and the usability of TRIZ-based computerized aid to inventive design.

Several attempts have been proposed in scientific literature about methods for determining why human-computer interaction does not produce the expected results (e.g: there is plenty of papers about self-efficacy studies, as well reviewed for gender differences in [5]). Hewett et al. in [11] presented a method for evaluating the performances of a computer tool supporting creativity. Nevertheless, nowadays, a standardized procedure that provides precise measurements about failures of computer-aided innovation systems in producing good results is still missing.

This paper proposes a set of general criteria to study the unsuccessful analyses of technical problems carried out by users of a TRIZ-based tool for problem solving, with the aim of identifying the sources of inefficiency and, consequently, the directions for improvement. A first exemplary application of these criteria is done through an in-depth investigation of the failures emerged during the tests carried out by engineers and industrial designers that used the above-mentioned OPEN-IT computer based system. The aim is to identify the main criticalities of the software application, so that it is possible to clarify if new requirements emerge and if those formally fulfilled have yet to be satisfied. Protocol Analysis [8] has been taken into account as a well established approach for inspecting design cognitive processes and characterizing mental paths and behaviours.

A brief description of the characteristics and of the capabilities of the algorithm is shown in the second section, together with contributions from literature that point out the attention to the role and requirements of computer applications within creative tasks. In addition, brief references about protocol analysis in design phases are mentioned. Section 3 clarifies the criteria for assessing the performance of a computerized tool for inventive design and the fourth section presents a detailed examination of the outcomes achieved through their application to the OPEN-IT experimental results and discusses about the main directions of development to be undertaken in order to improve the system. Eventually, Section 5 summarizes the original contribution of the paper and briefly discusses the evidences emerged in the analysis.

2 Computer-Aided Problem Solving: lessons learned from past experiences

As mentioned above, in order to improve the characteristics of the existing computerized systems for problem solving in design tasks, the authors proposed a computerizable algorithm for problem analysis, implemented in the OPEN-IT dialogue-based framework, whose characteristics are here briefly presented.

Insights from Scientific Literature Computer-aided systems for Problem Solving have to embed different characteristics [4]; some of those requirements were already discussed by other scholars. Lubart [14] recognize that the coaching of designers, acting as an expert system that guides the user throughout cognitive processes, is one of the utmost roles a computer can play. Hewett [10], from a different perspective, claimed that the problem analysis should be carried out by taking into account different facets of the problem, enlarging the range of investigation. Aurisicchio et al. [2] pointed out that the information gathering is a time consuming activity in design phases, concluding that it is necessary to ease the research of relevant contents from knowledge sources.

TRIZ-related requirements Consistently with TRIZ [1] and OTSM-TRIZ [7], a successful problem solving activity,capable to produce breakthroughs, is characterized by different aspects. The user should be supported throughout an abstraction process of problem features, as to focus just on the characteristics the solution should have until the convergence towards a unique and formalized description. Moreover, this abstraction activity should foster the user in defining technical barriers (in TRIZ terms contradictions) that prevent a direct implementation of typical solutions. At last, it is strictly required that such a system does not need the user to hold a long education period to become effective, in order to improve its usability in contexts where scarce resources for training courses are available.

OPEN-IT - Algorithm for Problem Analysis In order to embed all the abovementioned characteristics into a computer aided-system for problem solving, the authors built a dialogue-based algorithm for the analysis of technical problems, whose latest updates have been published in [3]. A full description of the algorithm is out of the scope of the present paper. In brief, the computerized procedure is composed by more than 200 nodes organized in eight logical blocks. The nodes represent an articulate set of questions, choices or written messages exploiting a common terminology, rather than TRIZ jargon. Several nodes are aimed at checking the correctness of previous user answers, so that they can be employed in contextualizing the text of the following questions. The ultimate objective of the procedure is identifying the most critical TRIZ contradiction behind a given problem.

As briefly mentioned in the introduction, all the TRIZ-based systems supporting inventive design suffer from poor efficacy [6], especially for those individuals who are not experienced in using abstract models. Despite the intention to go beyond the limits of current commercial systems, some failures have been recorded also by the authors in the testing campaign of the OPEN-IT framework.

Generally speaking, the existence of unsuccessful results may highlight that some requirements are just partially satisfied, either in terms of functionality or in terms of usability; alternatively, new requirements still need to be elicited. Thus, a relevant objective is the definition of a set of criteria capable to shed light on the sources of limited performance. In this paper, the proposed criteria are applied in combination with a protocol analysis approach [8], due to its suitability in examining designers' behaviour along the design process, [12].

3 Criteria for Examining the Results of a Computer-Aided Problem Solving Activity

A designer carries out his design activity by focusing on different problem features, such as the performances to be achieved, the drawbacks to avoid or the consumption of resources. In order to overcome problems, the problem definition strategy can be carried out at a more or less abstract level, thinking about structures and embodiments, exploitable physical principles, as well as requirements and goals. Traditionally, a design protocol analysis is carried out by processing each design step performed by the designer or the design team; in this case, since the study is dedicated to computer-aided tools supporting inventive design activities, the protocol analysis is focused on the interactions between the user and the software system. The below proposed criteria are aimed at classifying the steps of a design activity, regardless the Computer-Aided tool adopted.

Criteria for Strategy Assessment Computer-Aided systems for problem solving may leave a complete freedom or, on the other hand, force the user into a predefined set of steps or instruments to cope with. Therefore, these criteria have to take into account both the extreme situations, so to encompass all possible cases. Then, regardless of designers’ choice or input request by the computer, the steps can be classified according to the following set of six criteria:

- *Functional Requirement:* Human-computer interactions (HCI) related to the elicitation of the objectives to be achieved by a given technical system.
- *Behavioural Variable:* HCI focusing on the mechanisms (physical, chemical, geometrical,…) that allow a certain phenomenon to take place.
- *Structural characteristic:* HCI taking into account specific design variables that allow to leverage a physical principle.
- *Choice:* HCI concerning decisions which are made without a particular reference to the strategy and the path to solve a problem.
- *Communication:* HCI dedicated just to transfer the designer information about the progressing process.
- *Check:* HCI through which the computer asks the designer about the correctness of previous steps.

Criteria for Assessing the Focalization on Problem Features As seen before, a designer may focus on different aspects of a problem. The below defined criteria are aimed at classifying them.

- *General features:* HCI focused on features and characteristics that are not directly related to the problem, but generally refer to the technical system.
- *Removal of drawbacks:* HCI characterized by undesired consequences emerging during the functioning of the technical system.
- *Presence of conflicts:* HCI aimed at individuating the elements of the problem that prevent the elimination of drawbacks.
- *Improvement of performance:* HCI that take into account a better achievement of performances for which the technical system has been designed.
- *Requirements for system functioning:* HCI coping with the means that allow the technical system to properly work.
- *Broadening spectrum of investigation:* HCI addressed at helping the designer to avoid fixation by exploring alternatives from a wider perspective.

Criteria for Assessing the Kind of Errors A traditional design activity, e.g. by using trial-and-error, is characterized by mistakes, or useless solution attempts, that reduce the efficiency of the problem solving process. The criteria defined hereafter are aimed at making a distinction between those mistakes:

- *Content:* Mistakes due to misunderstandings about how the system works, wrong interpretations of the mechanisms causing the undesired phenomena, poor investigation of the problem due to neglected elements or effects.
- *Form:* This class is a residual of the first one. Such mistakes are typically characterized by wrong insertions due to language issues, disregard of the directions to follow, as recommended by the computer system or dictated by the principles of the employed design methodology, etc.

The above definitions are clarified by the examples reported in the following Section. The overall set of criteria allows the exploration of designers' reasoning path according to different perspectives; however, it is required to specify which is the investigation sequence to follow, so that the analysis can be repeated in different contexts. The authors suggest to:

1. Record all the steps of a problem solving activity by means of an appropriate method for protocol analysis (think-aloud or conversational, concurrent or intro/retrospective, combinations…);
2. Determine the characteristics of each step, both in terms of strategy and focus;
3. Evaluate the correctness of the steps, following their original sequence, maximally through objective criteria (e.g. for form mistakes) or by experts' assessment when required;
4. Determine, for each incorrect step, whether the error regards the content or the form.

The above-presented criteria can successfully describe the steps of a problem solving activity carried out with the support of a computer tool, as confirmed by several tests carried out on different software applications. In detail, the questionnaire of Innovation Workbench (www.ideationtriz.com) easily allows to record the designer's activity and all the provided answers can be classified with reference to the suggested criteria. They can be also used to classify the steps carried out with Invention Machine's Tech Optimizer (www.inventionmachine.com), as well as Southbeach Modeller (www.southbeachinc.com). The recording of those design steps requires the additional employment of software for logging keystrokes to accomplish the above step 1.

4 Insights from Unsuccessful Problem Analyses and Discussion

The authors have carried out a high number of tests with their problem solving algorithm, owning an intrinsic capability to record design steps. Therefore it is pos-

sible to examine a significant sample of problem solving session logs, by using the proposed criteria and investigation procedure.
A group of graduates, composed by both males and females holding a MS or a PhD in Mechanical Engineering or in Industrial Design, has been asked to face two technical problems [3] emerged in real industrial contexts by means of the OPEN-IT platform presented in Section 2. Testers’ competencies on systematic problem solving were almost completely absent, since just an individual of the sample claimed to have been submitted to 20 training hours in TRIZ. Each of them holds at least a First Certificate in English or higher. At the end of the testing session, 24 analyses were selected for the present study, because they showed at least one error in any node of the questioning procedure and did not result in particularly valuable outcomes within the scope of solving the encountered problem.

A list of exemplary errors related to several criteria among those defined in Section 3, is reported below (words in brackets refer to terms previously introduced by the designer):

- *Functional Requirement* Question: “Which technical function is carried out by the <calendaring system> in order to <apply a film on surface>? Use the infinitive form of the verb without "to" (i.e. keep ink, dry the clothes, deliver a box...)”; Answer: “air bubble”.
- *Behavioural Variable* Q: “Which is the undesired effect that arises in the system as a consequence of getting the satisfactory level of the <cleanliness of the frying pan>? Use a noun without the article or a verb in the -ing form (e.g.: high noise, overheating,...)”; A: “avoid rivets, introducing a new film”.
- *Removal of drawbacks* Q: “Which is the undesired effect that arises in the system? Use a noun without the article (e.g. noise, bone breaking, vibrations, obstructed view...)”; A: “limited effect”. (Note: the user does not specify what effect he is talking about).
- *Presence of conflicts* Q: “Do any bad consequences come out if you <increase> the <roughness of the external surface> of the <driving roller>?”; A: “No”. (Note: the user does not realize that there is an undesired consequence).
- *Content* Q: “Define the instant or the initial condition in which the <rivets> start to <fix the mutual position>. ”; A: “dirt”.
- *Form* Q: “Define the instant or the initial condition in which the <mechanical joint> starts/start to <clamping the handle>.”; A: “assembly process”. (Note: the user is referring to the whole time interval, instead of the initial instant as requested).

The following examination counts the revealed punctual mistakes, regardless the quality of the final outcomes of the questioning procedure. Its results will be presented in an aggregate form, so to highlight whether the system globally satisfies the requirements it has been designed for. As shown in Table 1, the main evidence refers to the highest percentage of mistakes occurring along the steps concerning the “General Features” of a technical system. Furthermore, Table 2 shows that the most severe difficulties arise when the designer is asked to specify aspects of the technical system that are strictly related to mechanisms and the physical principles that determine the presence of both desired and undesired effects.

Table 1 Steps where users have provided a wrong answer both in terms of content and form. The percentage of wrong steps has been calculated as the ratio between the number of mistakes for the specific feature and the overall number of errors

Feature under analysis	*% of wrong steps*
General Features	58,72%
Removal of Drawbacks	10,47%
Presence of Conflicts	8,72%
Improvement of Performances	8,72%
Requirements for System Functioning	0,00%
Broadening the Spectrum of Investigation	13,37%

Table 2 Summary of incorrectly answered steps. Communication and Check mistakes have been taken into account because they present errors caused by previous wrong steps.

Focus of the analysis	*% of wrong steps*
Functional Requirement	28,68%
Behavioural Variable	46,51%
Structural Characteristic	15,50%
Choice	1,55%
Communication	6,50%
Check	1,55%

In order to gather with greater accuracy information about the sort of encountered problems, the authors collected detailed insights about the marginal distribution of this kind of errors, characterizing them also in terms of "content" and "form" as shown in Table 3. In the Table, a further cluster of "mistakes" is illustrated, which is relevant for the specific CAI application adopted or for any dialogue-based system. The Table includes the counting of those queries for which the designer gave no answer.

Table 3 Correlations between strategy and kind of errors. Misalignments in sums are due to the rounding of percentages.

	Content	*Form*	*Not assigned*	
Functional	23,26%	5,43%	0,00%	28,68%
Behavioural	36,43%	10,08%	0,00%	46,51%
Structural	14,73%	0,78%	0,00%	15,50%
Choice	1,55%	0,00%	0,00%	1,55%
Communication	0,00%	0,00%	6,20%	6,50%
Check	0,00%	0,00%	1,55%	1,55%
	75,97%	16,28%	7,75%	100,00%

A noticeable presence of mistakes related to the form may highlight that the main observed limitations could be due to the user interface of the computer-aided system, rather than on the poor knowledge of the designer about the physics of the specific issue under investigation. Nevertheless, the results of Table 3 do not confirm such hypothesis, although further investigations are needed to obtain more robust indications. An in-depth analysis of the results shows that the algorithm still lacks in the capability of abstracting problem features. Indeed, according to the data illustrated in Table 2, the most critical aspects to be investigated are related to functional features and behavioural variables, while just a minor percentage of errors concern the structural characteristics of a technical system.

Moreover, it is worth to reflect upon the uneven distribution of feature related errors, since diverging conclusions may emerge according to different interpretations. On the one hand, designers could have encountered troubles since the beginning of the procedure, where “General Features” related questions are asked; on the other hand, the same testers could have paid less attention to those aspects that they do not consider essential for the description of the problem.

Therefore, the above considerations show how this preliminary criteria-based analysis is capable to reveal the main essence of flaws in computer-aided systems and allows to plan appropriate strategies to overcome the arisen shortcomings. A detailed examination of individuals’ behaviour may result as an important element to distinguish the facets that are still ambiguous, or, as well, to determine the extent of the factors that generate problems along the analysis of technical systems.

5 Conclusions

This work briefly summarizes the requirements of a Computer-Aided Problem Solving System. The authors propose an original metric to evaluate the behaviour of designers using this sort of tools, so that it is possible to analyze the steps of the design process according to definite criteria. The purpose is to start to examine why specific human-computer interactions produce failures, allowing to highlight whether the requirements already addressed by the literature are still far to be met and, if needed, to elicit new ones.

An application of this kind of protocol analysis has been conducted on 24 tests using a computer-aided system for problem analysis developed by the authors. The examination of the results in an aggregate fashion allows to make preliminary considerations about the degree of achievement of the above-mentioned requirements. The obtained results show the directions of development to be prioritized in order to improve the framework of the computer-aided system. Specifically, a relevant feature to be addressed concerns its capability to support the user in abstracting the problem and allowing him/her to focus with more attention on facets that may appear as marginally related to the problem, but that can hide potential direction for its solution.

Further insights about the analysis of the results obtained by the individuals could represent an interesting point to be discussed, especially considering the branch of protocol analysis in computer-aided design that is among the purposes of future investigations by the authors.

Eventually, this general approach can be also easily replicated on different computer-aided systems for problem solving that use a different way to structure the designers' knowledge. Related results can constitute a starting point to share a common vision on what should be done for obtaining a more mature and reliable computerized means for supporting the creative stages of the design process.

References

1. Altshuller GS (1984) Creativity as an exact science. Gordon and Breach Science Publisher, New York.
2. Aurisicchio M, Bracewell R, Wallace K (2006) Characterising the information requests of engineering designers. In: 9th International Design Conference.
3. Becattini N, Borgianni Y, Cascini G, Rotini F (2011) A TRIZ-based CAI Framework to guide Engineering Students towards a Broad-spectrum Investigation of Inventive Technical Problems, submitted for publication on the International Journal of Engineering Education, in press, 2011.
4. Becattini N, Borgianni Y, Cascini G, Rotini F (2011) Model and algorithm for computer-aided inventive problem analysis, Computer-Aided Design doi:10.1016/j.cad.2011.02.013.
5. Burnett MM, Beckwith L, Wiedenbeck S et al. (2011) Gender Pluralism in Problem-Solving Software, Interacting with Computers, doi:10.1016/j.intcom.2011.06.004.
6. Cavallucci D (2011) A research agenda for computing developments associated with innovation pipelines. Computers in Industry 4:377-383.
7. Cavallucci D, Khomenko N (2007) From TRIZ to OTSM-TRIZ: addressing complexity challenges in inventive design. International Journal of Product Development 4:4–21.
8. Ericsson KA, Simon HA (1993) Protocol analysis: Verbal reports as data (rev. ed.). The MIT Press, Cambridge.
9. Funke J, Frensch PA (2007) Complex problem solving: the European perspective. In: Jonassen DH, editor. Learning to solve complex scientific problems 25–47. Lawrence Erilbaum, New York.
10. Hewett T (2005) Informing the design of computer-based environments to support creativity. International Journal of Human–Computer Studies 63:383–409.
11. Hewett T, Czerwinski M, Terry M et al. (2005) Creativity support tool evaluation methods and metrics. Report of NSF workshop on creativity support tools (available at http://www.cs.umd.edu/hcil/CST/report.html), 10-24.
12. Hüsig S, Kohn S: (2011) Open CAI 2.0" - Computer Aided Innovation in the era of open innovation and Web 2.0. Computers in Industry 62(4): 407-413.
13. Jiang H, Yen CC, (2009). Protocol Analysis in Design Research: a review. Paper presented at the "Design | Rigor & Relevance", International Association of Societies of Design Research (IASDR) Conference, Seoul, Korea.
14. Lubart T (2005) How can computers be partners in the creative process: classification and commentary on the special issue. Int. Journal of Human–Computer Studies 63:365–9.
15. Simon HA (1973) The structure of ill-structured problems. Artificial Intelligence 4:181–201
16. Tomiyama T, Gu P, Jin Y et al. (2009) Design methodologies: Industrial and educational applications. CIRP Annals - Manufacturing Technology 58(2):543-565..